



## **1097 Editor interactiu de programes docents**

Memòria del Projecte Fi de Carrera  
d'Enginyeria en Informàtica  
realitzat per  
Javier Román López  
i dirigit per  
Lluís Ribas Xirgo  
Bellaterra, 17 de Setembre de 2009



# Taula de continguts

CAPÍTOL 1: INTRODUCCIÓ.....	5
1.1 CONTEXT .....	5
1.2 MOTIVACIÓ .....	6
1.3 OBJECTIUS .....	7
1.4 TASQUES .....	9
1.5 PLANIFICACIÓ .....	11
 CAPÍTOL 2: ESTAT DE L'ART .....	13
2.1 EDITORS COMERCIALS .....	13
2.2 EDITORS LLIURES .....	14
2.3 FORMATS PER REPRESENTAR GRAFS.....	16
 CAPÍTOL 3: DISSENY DE L'INTERFICIE D'USUARI.....	18
3.1 REQUISITS DE L'APLICACIÓ .....	18
3.1.1 REQUISITS FUNCIONALS .....	18
3.1.2 REQUISITS NO FUNCIONALS.....	19
3.2 FENOTIP/GENOTIP .....	19
3.3 ORDENACIÓ TOPOLOÈGICA.....	21
 CAPÍTOL 4: INTEROPERABILITAT AMB L'ENTORN DE VISUALITZACIÓ D'AGENDES .....	22
4.1 INTRODUCCIÓ .....	22
4.2 XML ENCODER I XML DECODER .....	23
4.3 INTEROPERABILITAT DE L'APLICACIÓ .....	23
4.4 QUÈ SÓN ELS DATAOBJECT.....	24
4.5 DESCRIPCIÓ DEL XML DE L'ENTORN D'AGENDES.....	25
 CAPÍTOL 5: APLICACIÓ.....	27
5.1 MENÚS I BARRES D'EINES .....	28
5.2 EXEMPLE DE CONNEXIÓ DELS NODES .....	30
5.3 EDITAR ELS CONTINGUTS DELS NODES .....	30
5.4 ORDENACIÓ TOPOLOÈGICA.....	31

CAPÍTOL 6: DESCRIPCIÓ TÈNICA .....	34
6.1 ESTRUCTURA INTERNA .....	34
6.2 EDITOR DE NODES .....	35
6.3 VISTA DELS NODES.....	36
6.4 ORDENACIÓ DEL GRAF.....	37
 CAPÍTOL 7: CONCLUSIONS .....	 38

# Capítol 1.

## Introducció

### 1.1 Context

A partir de la implantació del sistema European Credit Transfer System (ECTS) en l'àmbit de l'Espai Europeu d'Educació Superior (EEES), la nova metodologia docent, ha augmentat la importància de les guies docents. Això és degut a que l'alumne passa a ser l'actor principal del procés d'aprenentatge.

Aquest fet ha impulsat la creació de les *agendes de aprenentatge* [1]. Són llistes d'activitats d'aprenentatge que haurà de dur a terme l'alumne durant el curs, i que estan ordenades en el temps, amb la possibilitat de diferents maneres de visualització.

Les activitats proposades poden ser presencials o no presencials. Les primeres són classes magistrals, seminaris, sessions de laboratori, tutories o exàmens. En canvi, les no presencials recullen les hores d'estudi que hi dedica l'alumne, la resolució de problemes, la preparació dels seminaris, previs de pràctiques o documents que hagi d'entregar. Aquestes activitats han de comptar amb recursos perquè es puguin dur a terme, com ara aules i professors. També s'inclou com a recurs, el grup d'alumnes a qui va dirigida l'activitat. Habitualment, els grups de classe, però també els equips de pràctiques i els mateixos estudiants, de forma individual.

El fet de fer una sincronització entre el calendari i el projecte d'aprenentatge fa necessari l'ús d'alguna eina o aplicació ja existent per a la gestió de projectes. Tot i que aquestes eines permeten la generació de l'agenda, no són adequades per la representació de continguts. Això implica que només utilitzant eines per la gestió de projectes tindrem dificultats per posar a disposició dels alumnes les agendes d'activitats, per tant també

caldrà fer ús d'algun gestor de continguts (CMS, segons les sigles en anglès) que permeti poder tenir diferents visualitzacions als usuaris..

D'aquesta manera, es permet als alumnes poder tenir una visualització personalitzada de l'agenda d'activitats, on poden afegir restriccions o condicions de caràcter personal, canviar la visualització de algunes activitats com pot ser el color de representació per donar-li més importància o menys o fins i tot la manera com esta representada la successió d'activitats que l'alumne ha de completar per tal d'assolir els objectius fixats.

Tot això crea la necessitat de disposar d'alguna eina per automatitzar el procés d'elaboració de les agendes d'aprenentatge que sigui fàcil d'utilitzar, adaptable a qualsevol tipus d'assignatura i ens ofereixi les capacitats necessàries per tractar els nostres continguts ja que encara que les guies docents es poden editar fent servir software genèric de planificació o d'edició de calendaris d'activitats, en molts casos no ofereixen totes les funcionalitats que necessitem i són difícils de modificar per adaptar-los a les nostres necessitats. També existeixen un tipus d'aplicacions anomenades de gestió de projectes però ja es va comprovar fa uns anys per part del director d'aquest projecte que també eren difícils d'adaptar.

Tot el que s'ha esmentat juntament amb la necessitat d'un software intuïtiu, que ens ofereixi tot el que necessiten els professors, però sense ser molt complicat ja que l'objectiu és ser un editor de guies de docents interactiu, gràfic i via web.

## **1.2 Motivació**

Està clara la necessitat d'una eina per a elaborar les guies o agendes d'aprenentatge. Per tant, aquesta aplicació, que ens permetrà introduir les guies docents de manera que tinguem una integració entre agenda i guia docent, serà de gran utilitat per poder generar automàticament les agendes d'aprenentatge que es mostraran als alumnes. Aquesta integració es durà a terme fora de l'àmbit de la aplicació, a partir de la informació de la guia docent que proporciona com a sortida l'aplicació i que permetrà crear les agendes d'activitats.

Sí que és cert que la informació que hi hagi a l'agenda es podrà formatar de manera que es pugui incloure a una guia docent o tractar automàticament per incloure-hi informació de la sincronització entre activitats i calendari acadèmic.

Actualment existeixen aplicacions que intenten solucionar aquesta mancança, encara que són incompletes com en el cas de la UOC [2]. Aquesta Universitat presenta en el seu campus virtual per una banda el calendari d'activitats i per una altra l'agenda. Un altre exemple el podem trobar a la pàgina de l'ETSE [3] de la UAB [4] on es pot tenir accés a les guies docents seguint el model del sistema de ECTS, però sense estar associat a un calendari.

Un altre aplicatiu el trobem al campus virtual de la UAB [5], que presenta un calendari on es poden situar totes les activitats de l'assignatura, però la creació d'aquest és manual.

A més d'aquesta manca d'integració entre agenda i guia, no hi ha cap model de dades que mantingui la separació entre la guia d'activitats d'aprenentatge i els recursos associats a cada tasca d'aquesta guia d'activitats.

## **1.3 Objectius**

L'objectiu principal del projecte és crear una eina per introduir les guies d'activitats d'aprenentatge de manera gràfica a partir de la informació de les guies docents i de la institucional en relació als estudis com, per exemple, horaris i espais. S'ha de garantir la interoperabilitat amb l'entorn actual de visualització d'agendes d'aprenentatge. És a dir, introduir l'agenda d'activitats que l'alumne haurà de dur a terme durant el curs. Aquestes activitats tenen unes relacions de precedència, normalment per continguts i tipus d'activitat (les presentacions de continguts han de precedir la realització de problemes, per exemple), que s'han de respectar. Per tant, la idea és tractar aquestes

guies d'activitats d'aprenentatge com un graf, on els nodes són les activitats i els arcs són les dependències.

D'altra banda, la sortida de l'editor de guies docents ha de ser compatible amb l'aplicació que actualment s'utilitza per mostrar aquestes guies d'aprenentatge als alumnes mitjançant el campus virtual de la UAB.

En tractar-se en definitiva d'un editor de grafs, l'aplicació ha de permetre la manipulació d'aquests grafs, oferint diferents representacions de les tasques que componen la guia docent i mostrant canvis morfològics en funció del contingut del node, reordenar-los de manera que es presentin visualment millor estructurades i ha de permetre per a cadascuna d'aquestes tasques introduir el títol, codi, descripció, lloc, data i hora, professor, enllaços a recursos en línia, bibliografia, comentaris, etc... i permetre l'exportació de l'estructura del graf a formats com XML [7].

Hi ha eines com yED [6] que proporcionen un funcionament molt similar al que s'intenta arribar en aquest projecte. L'inconvenient d'aquesta eina és que és molt potent i el seu ús es bastant complicat per a persones que no estan habituades a treballar amb ordinadors. Existeixen moltes més eines de les que es parlarà al capítol 2.

Per arribar a aquest objectiu s'utilitzarà la biblioteca de grafs JGraph. És una eina molt potent que ens permet, de forma fàcil, la representació de grafs. Al capítol dos es parlarà d'aquesta eina.



## 1.4 Tasques

Reunir-se amb el director del projecte.

Reunió el de 10/11/08 on es va assignar el projecte i es va fer l'explicació del mateix.

Documentació.

Es tracta de fer un estudi de l'art per tal de trobar la millor solució al problema que es planteja en aquest projecte. Documentació sobre tipus d'editors, eines per tractar grafs, tipus de dades que es fan servir, etc...

Total: 40 hores

Desenvolupament i cerca d'informació.

Un cop fet l'estudi de l'art i trobada una possible solució, s'ha de fer un estudi i una cerca d'informació sobre la llibreria JGraph, la representació de grafs en XML i les solucions per crear interfícies gràfiques que proporciona Java.

Total: 10 hores

Disseny de la interfície gràfica.

El projecte que es vol desenvolupar es tracta d'una eina que faciliti la introducció de les guies docents, per tant serà una eina que té com a requeriment una interfície gràfica per introduir i interactuar amb els grafs.

Total: 100 hores

Desenvolupament de la part de manipulació de grafs mitjançant la llibreria JGraph.

Un cop feta la interfície gràfica, la següent tasca fa referència a la part d'introduir les dades, la manipulació d'aquestes i les diferents representacions i canvis morfològics que aquestes poden tenir.

Total: 60 hores

Disseny i programació d'una aplicació web per accedir al editor.

Un cop finalitzada la part de l'editor de guies docents, s'ha de fer la segona part d'aquest projecte, l'accés a l'editor per tal de crear, manipular o visualitzar en el cas dels alumnes. Això es farà mitjançant una aplicació web.

Total: 50 hores

Mòdul per exportar les dades a XML.

Un dels requeriments del projecte és la sortida de les dades en format XML, per tant s'haurà de programar un mòdul que permeti l'exportació des de GXL al format XML que s'utilitza en el visualitzador de guies docents implementat en proves al Campus Virtual de la UAB.

Total: 20 hores

Programació d'un mòdul per convertir les dades XML del format GXL a SVG.

La llibreria JGraph fa servir una variant de XML anomenada GXL per representar els grafs, per tal de estandarditzar aquesta representació serà necessària una conversió a SVG amb el que tindrem un tractament intern del graf en format XML i a més estàndard per a futures ampliacions.

Total: 30 hores

Redacció de l'informe previ.

L'informe previ és, de fet, una primera versió del primer capítol de la memòria corresponent.

Total: 20 hores

Redacció de la Memòria final.

S'anirà fent progressivament al final de cada tasca i inclou una etapa de revisió final. També s'hi inclou la preparació de la presentació del projecte.

Total: 50 hores

Prova de casos de test.

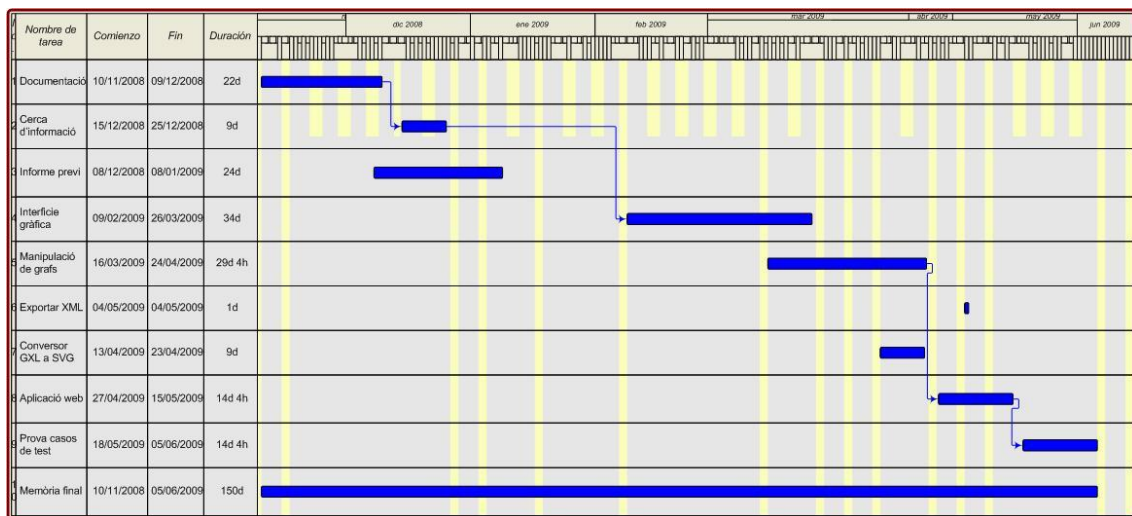
Per tal de garantir una bona funcionalitat s'hauran de fer tests.

Total: 40 hores

## 1.5 Planificació

A partir de les tasques anteriors es pot fer una planificació on es pot veure que en principi amb un total de 430 hores es podrà realitzar el projecte. És un treball de 450 hores, així que es té un total de 20 hores per poder ajustar la planificació per tal d'acabar el projecte en el termini fixat, al juny de 2009.

A continuació es pot veure un diagrama de Gantt on es poden veure les tasques a realitzar amb les precedències que tenen i amb el cost total en hores que representa cada tasca.



## Referències

- [1] Lluís Ribas Xirgo, Josep Velasco González, “La agenda de aprendizaje como herramienta de adaptación a la metodología educativa en ECTS del EEES”, ISBN: 978-84-9732-620-9, Thomson, Jul.07
- [2] Campus Virtual de la UOC. [<http://www.uoc.edu/presentacions/campus>]
- [3] Pàgina web de l’Escola Tècnica Superior d’Enginyeria de la UAB, ETSE. [<http://www.etse.uab.cat>]
- [4] Universitat Autònoma de Barcelona, UAB. [<http://www.uab.cat>]
- [5] Campus virtual, UAB. [<http://cv.uab.cat>]
- [6] “yED Graph Editor”, yWorks, 28/12/2008. [<http://www.yworks.com>]
- [7] “XML”, 28/12/2008. [<http://www.w3.org/XML/>]

# Capítol 2.

## Estat de l'art

Durant la documentació prèvia una de les tasques va ser la de fer un estudi de les eines que hi havia disponibles fins al moment, tant comercials com lliures per tal de decidir de quina manera es podria realitzar l'editor i amb quines eines. Fixar-se en eines comercials té per objectiu prendre idees per a la part de la interfície gràfica, ja que es vol crear un software que sigui fàcil i familiar per als usuaris.

### 2.1 Editors comercials

Un dels principals editors comercials es Microsoft Visio [1], que és un software de dibuix vectorial per a Microsoft Windows. Les eines que el componen permeten realitzar diagrames de bases de dades, diagrames de flux, diagrames UML, etc... Existeix un software que intenta ser l'alternativa lliure a Visio anomenat Dia, del qual es parlarà més endavant.

Uns altres editors comercials encara que més accessibles al Microsoft Visio són el ConceptDraw [2] que és un software de dibuix vectorial que permet importar i exportar a XML i el SmartDraw [3].

Un altre editor comercial es el JGraphpad Pro [4]. Aquest es basa en la seva llibreria lliure JGraph però incorpora més funcions com el posicionament automàtic del graf, exportació a XML, funció de “drag & drop” per modificar els nodes i possibilitat de fer zoom i tenir una vista global.

Com ja s'ha comentat al capítol 1 existeix un editor via web que presenta unes característiques molt semblants a les que es volen aconseguir en aquest projecte. Es tracta de l'editor yED, aquest es pot utilitzar lliurement, però també existeix yFiles [5]. Aquest existeix en 3 llenguatges: Java, .NET i web 2.0. És una llibreria que proporciona algorismes per l'anàlisi, visualització i el posicionament automàtic del graf, diagrames o xarxes.

## 2.2 Editors lliures

Dins de l'apartat d'editors lliures parlarem d'editors gràfics i d'eines de visualització com poden ser llibreries que permeten la representació dels grafs per integrar-les en una aplicació o d'altres que a partir d'una descripció del graf en XML creen la representació del graf.

Com a editors trobem el ja comentat yED, aquest és d'ús lliure però el seu codi no està alliberat. D'altres com és el cas de GraphViz [6] és de codi lliure (Llicència CPL). Es tracta principalment d'un visualitzador que agafa la descripció del graf a partir d'un llenguatge molt simple i pot exportar el graf a diferents formats molt útils com imatges, format SVG [7] per a la web, Postscript, GXL [8] i XML. D'aquests formats de representació es parlarà més endavant al apartat anomenat Formats per representar grafs d'aquest mateix capítol.

Un altre editor és Dia [9] , inspirat en el software comercial Microsoft Visio utilitza un format propi basat en XML per carregar i guardar els diagrames i el seu codi font està programat en C++. Permet afegir nous tipus de nodes a partir de fitxer en XML.

Deixant de banda els editors hi ha moltes llibreries molt interessants, la majoria estan programades en Java i acompanyades de la seva API com és el cas de la llibreria

JGraph. Aquesta llibreria d'ús lliure s'integra com una subclasse del component Swing, el que implica una integració molt fàcil dins de l'aplicació final. Aquesta eina permet carregar grafs en XML i exportar-los a més de tenir funcionalitats típiques dels editor gràfics com zoom, desfer l'última acció, duplicar nodes, etc..

És una llibreria molt utilitzada, cosa que ha provocat que s'hagi creat noves versions o modificacions com es el cas de la llibreria JGraphT [10], principalment es basa en la mateixa llibreria però incorpora algunes característiques que depenent de la finalitat del software poden ser molt útils. Algunes d'aquestes característiques són la preparació per al tractament de grafs molt grans, la incorporació d'algoritmes matemàtics per tractar els nodes a més de estar dissenyat per donar altes prestacions. Fa ús de la llibreria Jgraph i a més les dues es poden utilitzar conjuntament.

Existeixen altres llibreries como es el cas de GVF (The Graph Visualization Framework) [11] que té llicència GPL, presenta característiques similars als anteriors, i fa servir GVF per la representació dels grafs. Existeixen d'altres com JUNG [12] o TouchGraph Graph Layout [13].

En aquest projecte es farà servir la llibreria JGraph, encara que en un principi sembli que l'ús d'aquesta llibreria portarà a fer més feina de programació, tenint en compte totes les característiques que presenta abans citades es pot veure que la programació es centrarà en l'entorn gràfic i bàsicament en implementar el requeriments del projecte, ja que la resta de funcionalitats que ha de presentar l'editor ja estan implementades. Per altra banda una de les raons que provoca l'ús d'aquesta llibreria es la dificultat d'adaptar un software encara que de codi obert i lliure com Dia les nostres necessitats.

A més, JGraph està programat en el llenguatge Java, el que permet un software multi plataforma i de fàcil accés per via web.

## 2.3 Formats per representar grafs

Existeixen diferents formats estàndards per la definició de grafs, aquests estan basats en XML ja que es tracta d'un llenguatge simple però molt potent a l'hora de descriure formats de dades.

El format que fa servir la llibreria JGraph es el GXL (Graph eXchange Language), aquest va ser presentat l'any 2000 i va ser dissenyat per permetre la interoperabilitat entre software com parsers, analitzadors i visualitzadors. El llenguatge GXL te dues propietats diferents a la resta com són, el model conceptual de dades és tipat, amb atributs i és un graf dirigit. L'altre propietat és que pot ser utilitzat per representar instàncies de dades com a esquemes per descriure l'estructura de les dades.

Per altra banda trobem el format SVG que és un llenguatge per descriure gràfics vectorials bidimensionals, encara que aquest no està dissenyat específicament per a la representació de grafs, sí que és possible representar-los. El gran avantatge que presenta SVG és que es molt potent, permet imatges animades i està incorporat de manera nativa a la majoria de navegadors webs que existeixen.



## Referències

- [1] “Microsoft Visio”, 07/01/2009,  
[<http://office.microsoft.com/es-es/visio/FX100487863082.aspx>]
- [2] “ConceptDraw”, 07/01/2009, [<http://www.conceptdraw.com>]
- [3] “SmartDraw”, 07/01/2009, [<http://www.smartdraw.com/>]
- [4] “JGraphPad”, 07/01/2009, [<http://www.jgraph.com/jgraphpad.html>]
- [5] “yFiles”, 07/01/2009, [[http://www.yworks.com/en/products\\_yfiles\\_about.html](http://www.yworks.com/en/products_yfiles_about.html)]
- [6] “GraphViz”, 07/01/2009, [<http://www.graphviz.org/>]
- [7] “SVG”, 07/01/2009, [<http://www.w3.org/Graphics/SVG/>]
- [8] “GXL”, 07/01/2009, [<http://www.gupro.de/GXL/>]
- [9] “Dia”, 07/01/2009, [<http://projects.gnome.org/dia/>]
- [10] “JGraphT”, 07/01/2009, [<http://jgrapht.sourceforge.net/>]
- [11] “GVF”, 07/01/2009, [<http://gvf.sourceforge.net/>]
- [12] “JUNG”, 07/01/2009, [<http://jung.sourceforge.net/>]
- [13] “TouchGraph”, 07/01/2009, [<http://sourceforge.net/projects/touchgraph>]

# Capítol 3.

## Disseny de la interfície d'usuari

### 3.1 Requisits de l'aplicació

A l'inici del projecte es van decidir una sèrie de requisits i objectius relacionats amb la interfície gràfica per tal d'aconseguir un software que s'adaptés a les necessitats de tenir una aplicació fàcil d'utilitzar i amigable, a més de complir amb la necessitat de tenir una sortida de les dades que siguin interoperables amb altres aplicacions de sincronització de calendaris.

#### 3.1.1 Requisits funcionals

La majoria de requisits del projecte són de tipus funcional, ja que estan relacionat amb la interfície d'usuari i com aquest interactua amb l'aplicació.

Un dels requisits funcionals és tenir nodes amb una estructura de dades determinada i a més la possibilitat de reservar recursos per cadascuna de les activitats o nodes. Això ja provoca que tinguem una aplicació adaptada a les nostres necessitats però no la farà molt diferent de la resta de les aplicacions existents al mercat, per tal d'aconseguir una aplicació diferent un altre dels requisits funcionals és que els nodes tinguin diferents fenotips a partir dels seus genotips, es a dir, que els nodes puguin canviar depenent dels recursos que tinguin assignats, de si tenen grups d'alumnes assignats o no, etc...

Finalment dintre dels requisits funcionals principals trobem l'ordenació topològica. Aquesta consisteix en ordenar el graf en forma d'arbre evitant que el arc es creuin o passin per sobre dels nodes. L'ordenació topològica es durà a terme mitjançant la petició de l'usuari i no de forma automàtica.

### **3.1.2 Requisits no funcionals**

Els requisits no funcionals fan referència a les restriccions del sistema com per exemple la necessitat de que les dades es puguin exportar en format XML i a més fent servir una estructura i una sintaxi específica. Sobre aquest requisit es parlarà al capítol 4.

Un altre requisit és el tenir un software que pugui ser accessible des de qualsevol ordinador i a més hauria de ser accessible des de internet, això provoca que una de les solucions més factibles sigui la de programar l'aplicació mitjançant el llenguatge de programació Java.

## **3.2 Fenotip/Genotip**

Un dels elements importants és la inclusió de fenotips al nodes, els nodes en un principi tenen una estructura de dades en la que li podem assignar el nom de l'activitat, el tipus, la duració de l'activitat, el títol, una descripció i si fos necessari, una url amb documentació adicional per a aquesta activitat.

D'altra banda, podem assignar grups d'alumnes a aquestes activitats, sempre que es parla d'activitat es refereix als nodes, i aquests grups poden tenir informació com per exemple el nom del grup, l'aula on es durà a terme aquesta activitat per part d'aquest grup o un professor responsable del grup.

Depenent d'aquesta informació o recursos assignats a les activitats, els nodes tindran diferents fenotips, és a dir, que canviaran de forma depenent de la informació que continguin. Els nodes poden tenir fins a 4 fenotips diferents, un de bàsic per a la

descripció de l'activitat com el de la figura 4, si li assignem un grup o més a aquesta activitat el node canviarà automàticament i tindrà la forma de la figura 2, i un altre per activitats amb recursos (a part dels alumnes) associats que tindrà la forma de la figura 3 i finalment si té recursos relacionats amb l'activitat i a més te assignat grups o reserva de recursos per a aquests grups el seu fenotip serà com el de la figura 1.

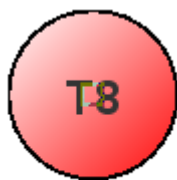


Figura 1



Figura 2

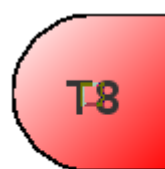


Figura 3



Figura 4

Així, els nodes que tinguin una forma com la de la figura 1 serien activitats que es poden dur a terme, mentre que les que tenen alguna vora recta no. Llavors podem tenir 4 formes diferents per representar els nodes, però això no vol dir que nosaltres podem triar la representació que volem tenir dels nodes a priori, ja que això hi han moltes aplicacions que ho permeten i no és l'objectiu d'aquest projecte.

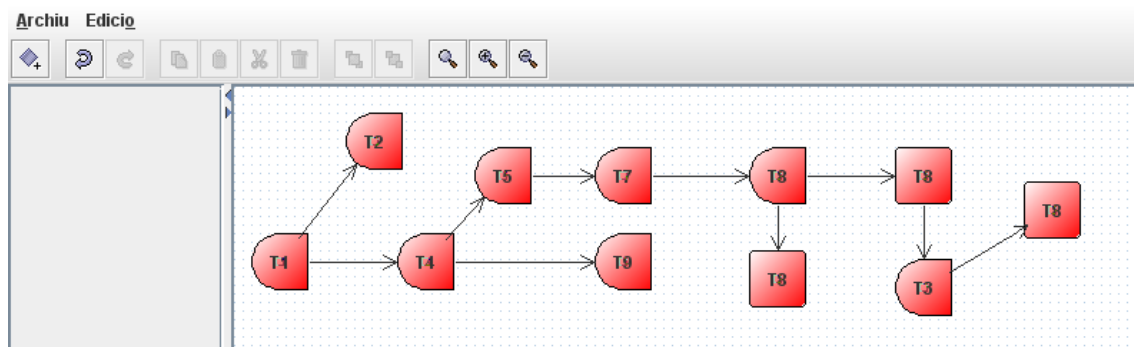
L'objectiu que s'intenta aconseguir és poder tenir una representació del contingut visualment i que aquest canviï al mateix temps que el contingut canvia. Per tant simplement observant el graf generat podem saber quines tasques tenen uns certs recursos assignats o si tenen diferents grups d'alumnes, etc...

La llibreria JGraph per representar la vista dels nodes fa servir el que anomenen VertexRenderers, per tal d'implementar el canvi de fenotip en un primer moment es va optar per fer servir un renderer específic per cadascuna de les formes, però el treballar amb diferents objectes era molt més complicat que fer servir un renderer universal a l'hora d'actualitzar la vista. Per tant, internament el funcionament és el següent, en primer lloc, es guarda el tipus de representació en una variable del node com si fos una propietat visual més com el color, tamany, posició, etc... Com ja s'ha comentat abans depenent del contingut tindrem una representació o una altra, llavors s'actualitza la propietat que fa referència al tipus de representació i el renderer, mitjançant un switch representarà una forma o una altra.

### 3.3 Ordenació topològica

Dintre de l'editor de grafs un dels conceptes més importants és el de poder tenir un accés a les dades de forma visual, com s'ha parlat abans, existeixen uns nodes amb unes característiques determinades adaptades a les nostres necessitats i mitjançant el canvi de fenotip podem saber només amb la seva representació gràfica quina informació pot contenir el node o quines propietats estan assignades.

Quan el graf comença a ser prou gran com per poder representar una guia docent, que seria ja un cas real, aquesta facilitat per poder veure visualment la representació de la informació és cada cop més petita, sempre hi poden haver arcs que es creuin, o per exemple es pot donar el cas que es vulgui afegir-hi una nova activitat en un lloc que ja s'havia dissenyat amb anterioritat degut a canvis en la guia docent o simplement per oblit i ens trobem en el cas de que no hi ha espai com a la figura 5, llavors l'edició resultaria molt costosa.



Per tal d'evitar tots aquests problemes l'aplicació permet fer canvis topològics, ordenant els nodes seguint un determinat algorisme. Això vol dir què, no es tracta d'un sistema de validació per evitar creuament d'arcs entre nodes sinó que a més de evitar aquests creuaments, permet ordenar tot el graf o només els nodes que es tinguin seleccionats seguint un determinat algorisme aconseguint diferents representacions com la representació en forma d'arbre, representació radial amb un node central o fins i tot intentant aconseguir sempre una representació lineal.

# Capítol 4.

## Interoperabilitat amb l'entorn de visualització d'agendes

### 4.1 Introducció

Com ja hem parlat en capítol 1 i 2 l'ús de XML, és un requisit del projecte per tal de garantir l'interoperabilitat entre aquesta aplicació i d'altres relacionades amb la creació o gestió de les guies docents. Per altra banda també es farà servir XML per treballar amb les dades internes de d'aplicació i que no tenen res a veure amb l'informació que es necessita per la guia docent com poden ser les classes Java de cadascun dels components, la posició en l'espai de cada node, el color, forma, tamany, etc...

Per tant per una banda tenim que hem de generar un XML que ens permeti tenir una aplicació interoperable però també hem de tenir unes altres dades necessàries per al funcionament de l'aplicació i aquestes no són útils per les guies docents. Per solucionar aquest problema el que farem serà tenir dos mètodes per generar XML, un més genèric fent servir el XML encode de Java i un altre més específic programat per a les nostres necessitats.

## 4.2 XML Encoder i XML Decoder

Per tal de poder treballar amb els grafs és necessari una eina per tal de guardar i recuperar les dades. Això ho farem mitjançant l' XML Encoder de la classe de Java `java.beans.XMLEncoder`.

Java utilitza un mecanisme de delegacions persistents per identificar que informació de certes classes necessiten ser serialitzades. Una bona guia on es pot veure com funciona el mecanisme per fer persistents certes classes és l'article de Sun, *Using XML Encoder* [1], on s'explica com utilitzar l'XML Encoder i com fer ús de les delegacions persistents.

La idea bàsica és que es poden fer persistents les classes que necessitem per poder recuperar el graf quan cridem al XMLDecoder més tard. Així podem recuperar tots els objectes del graf de forma fàcil i sense la necessitat de programar tot aquesta part encara que l'XML generalitzat no serà interoperable així que es presenta la necessitat de generar un altre XML que només contingui els dataobject de cada node. Aquest altre mòdul per generar l'XML referent a les propietats dels nodes s'explicarà més endavant.

Finalment un cop tinguem l'XML generat per l'XMLEncoder per tornar a recuperar el graf original només li hem de passar el fitxer a l'XMLDecoder.

## 4.3 Interoperabilitat de l'aplicació

Un dels objectius principals de l'aplicació és que sigui interoperable, o sigui que les dades que generem les podem tractar amb una altra aplicació com per exemple podria ser un visor de guies docents. Per aconseguir això ja hem dit que fer servir el fitxer generat amb l'XMLEncoder de Java no ens serveix per tant ha d'haver una part que pugui generar aquest XML.

Les dades que realment interessin són tipus d'activitat, la durada de cada activitat, la descripció, si per exemple tenim un enllaç amb material relacionat, l'aula, els grups, el

professor responsable i tot això està emmagatzemat al dataobject de cada node. Per tant el codi XML generat només contindrà els dataobjects.

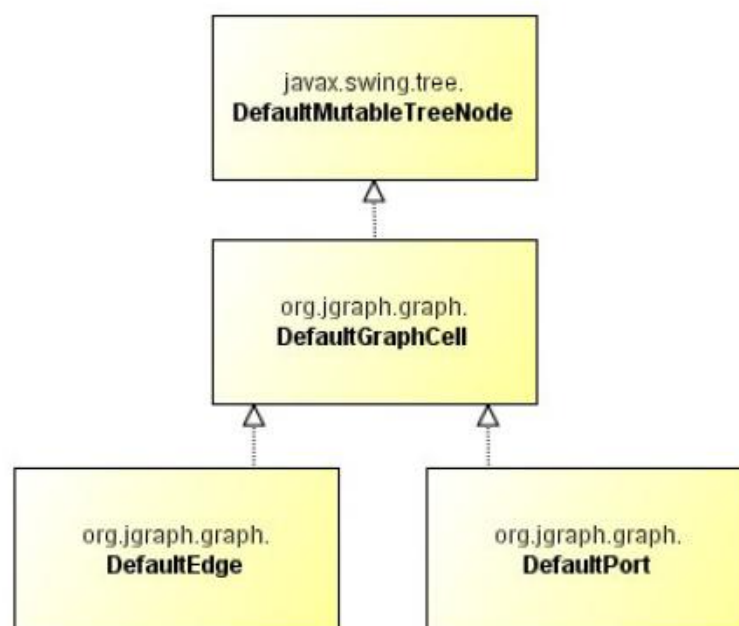
Per tant hi hauran dos “encoders” un més general que ens permet guardar objectes Java molt fàcilment però que no serveix per aconseguir l’objectiu de l’interoperabilitat i un altre més general que si ho aconsegueix però es necessària una programació extra.

## 4.4 Què són els DataObject

Els objectes principals llavors seran els vèrtexs del graf que són els que representen cadascuna de les activitats que l’alumne haurà de dur a terme durant el curs.

Aquests nodes pertanyen a la classe DefaultGraphCell que a la vegada és una extensió de la classe de Java DefaultMutableTreeNode [2], aquesta classe en el seu constructor rep un objecte que pot ser des de un String, el qual serà l’etiqueta del node fins a un objecte d’una classe com és en el cas d’aquest projecte.

La jerarquia de les classes dels vèrtexs, arestes i ports, es pot veure en el següent gràfic.





L'objecte que li passem al constructor de la classe DefaultGraphCell pertany a la classe NodeData, aquesta classe és la més relacionada amb la part del XML ja que XML inteoperable que hem de generar és una transcripció a XML de cadascun dels objectes d'aquesta classe que representa l'estructura d'una activitat.

## 4.5 Descripció del XML de l'entorn d'agendes

L'estructura del XML que hem de generar per tal de ser interoperable amb l'entorn d'agendes ha de ser com la següent. A l'exemple només tenim una activity, al fitxer en tindrem tantes amb aquest tag com nodes tingui el graf, cadascuna amb la seva informació. El tag allocation correspon als grups d'alumnes assignats a aquesta activitat, tindrem tantes allocations com grups hi hagin assignats a l'activitat, encara que no tinguin recursos assignats també han d'aparèixer.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<course>
  <subject id="21336">
    <assig>Sistemes Digitals II</assig>
    <activities>
      <activity id="T0a">
        <type>Teoria</type>
        <duration>60</duration>
        <title>Planificació de l'assignatura</title>
        <descr>Descripció de l'assignatura</descr>
        <url>Guia docent i planificacio.pdf</url>
        <allocation id="R10_T00">
          <alumni>G10</alumni>
          <begintime>
            <hour>14</hour>
            <min>00</min>
            <day>17</day>
            <month>1</month>
            <year>2009</year>
          </begintime>
          <endtime>
            <hour>14</hour>
            <min>05</min>
            <day>17</day>
```

```

        <month>1</month>
        <year>2009</year>
    </endtime>
    <location>--</location>
    <teacher>Mercè Rullán</teacher>
</allocation>
</activity>
</activities>
</subject>
</course>

```

Per tenir una aplicació compatible amb el visualitzador d'agendes l'aplicació ha de generar un fitxer amb una estructura semblant a la de l'exemple, aquest fitxer serà l'entrada del visualitzador, aquest el que farà es representar mitjançant una taula tota la informació que conte el xml sincronitzant la informació amb un calendari.

## Referències

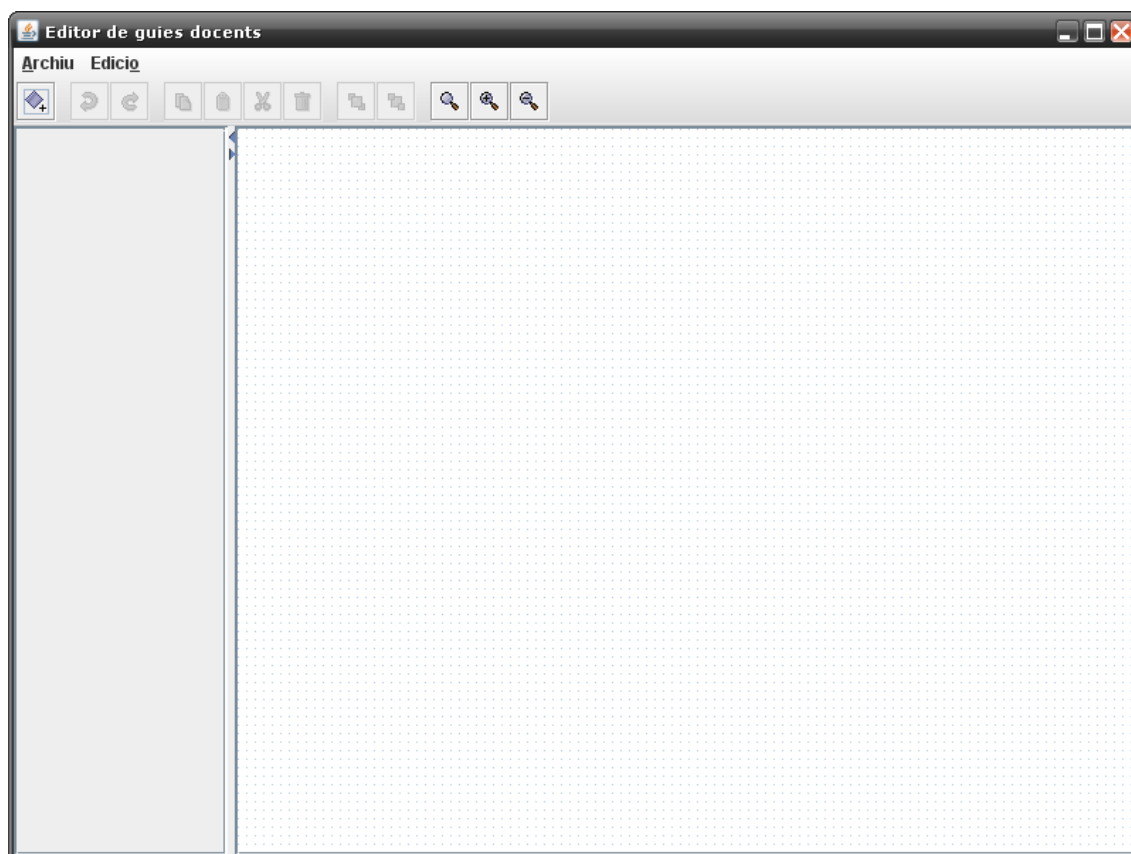
- [1] “*Using XML Encoder*”, 28/07/2009,  
[<http://java.sun.com/products/jfc/tsc/articles/persistence4/>]
- [2] API Java, 28/07/2009, [<http://java.sun.com/j2se/1.4.2/docs/api/javafx/swing/tree/DefaultMutableTreeNode.html>]

# Capítol 5.

## Aplicació

Aquest capítol intentar ser una mena de manual d'utilització de l'aplicació, per tal de mostra el seu funcionament mitjançant exemples i explicant de manera detallada les funcionalitats que ofereix.

La finestra principal del programa és la següent:

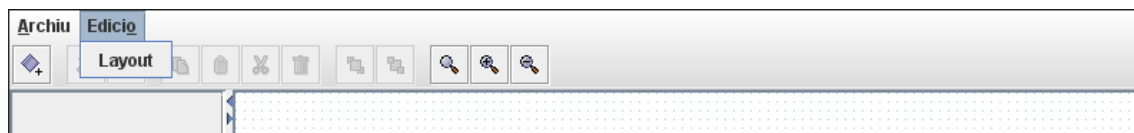
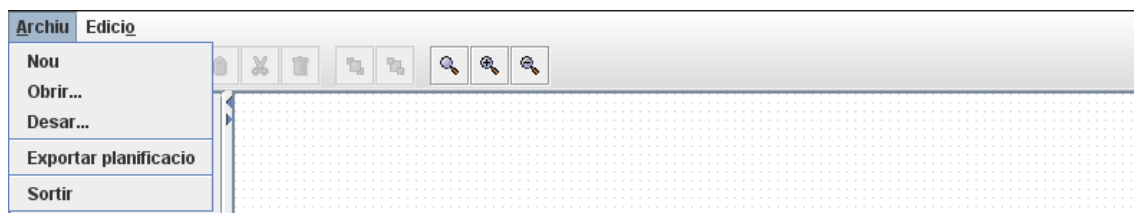


L'aplicació esta composta per un menú d'opcions i un altre d'eines, a més hi ha dos espais de treball, un espai a l'esquerra reservat per futures ampliacions com una llibreria

de nodes o una finestra amb una visió global del graf. L'altre espai, el de la dreta es l'espai de treball on es farà el disseny de les guies docents.

## 5.1 Menús i barres d'eines

Dintre del menú trobem dues opcions, Archiu i Edició, aquestes dues opcions ens permeten crear un document nou, obrir un projecte guardar anteriorment o guardar el treball actual a més de poder exportar el xml interoperable per el visualitzador de guies docents. L'altre opció ens permet modificar el layout, es a dir, ordenar el graf per evitar per exemple creuaments en els arcs. A continuació es poden veure els menús disponibles a l'aplicació.



Dintre de la barra d'eines hi ha diferents icones, a continuació hi ha una descripció del funcionament de cada icona.



Aquesta icona serveix per crear un nou node que i s'insertarà automàticament a l'espai de treball.



Aquestes dues icones serveixen per desfer i refer les últimes accions.



Aquesta icona serveix per copiar els nodes que hi hagin seleccionats en aquest moment.



Aquesta icona serveix per enganxar els nodes que hem copiat prèviament.



Aquesta icona serveix per retallar els nodes seleccionats, es guarden i poden ser enganxats més tard amb la icona de enganxar.



Aquesta icona serveix per esborrar nodes al igual que la tecla suprimir del teclat.



Aquestes dues icones ens permeten enviar un node al fons de l'espai de treball i llavors tots els altres nodes es mostraran sobre d'aquest o poden fer l'acció contrària, enviar el node al primer pla.



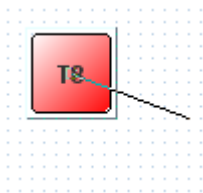
Aquestes icones serveixen per fer zoom, allunyant o apropant la vista o per retornar a la vista per defecte.

## 5.2 Exemple de connexió dels nodes

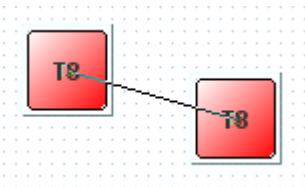
Per insertar nodes ja hem vist que es fa mitjançant la icona del rombe, llavors tindrem a la finestra un node com el següent:



Si continuem afegint nodes, el següent pas serà crear les relacions entre ells, per tal de fer això, es farà un clic al centre del node i sense deixar de clicar movent el cursor, al voltant del node apareixerà una mena d'envoltori. Això significa que el port del node està seleccionat i apareixerà a més un arc que segueix el cursor com es veu a la següent figura.



Per fer la connexió amb un altre node sense deixar de clicar es posa el cursor a sobre del node que volem que sigui a l'altre extrem de l'arc i a continuació aquest node passarà a tenir el seu port seleccionat, o sigui, amb l'envoltori blanc indicant la selecció.



## 5.3 Editar els continguts dels nodes

Una cop tenim un node insertar el que ens interessa és poder modificar els seus continguts, per poder fer això només hem de fer doble clic al node i a continuació apareixerà la finestra de l'editor de nodes. En aquesta finestra podrem modificar tota la informació relacionada amb les activitats a més de poder associar aquesta activitat a un grup o varis grups d'alumes.

The screenshot shows a window titled "Edit T8" with two main sections: "Activitat" and "Allocation".

**Activitat Section:**

- Activitat: T8
- Tipus: Teoria
- Duracio: 60
- Titol: Node de prova
- Descripcio: -
- Url: -

**Allocation Section:**

- A dropdown menu shows "R1\_T8".
- Buttons: "Afegir" and "Eliminar".
- Fields:
  - Allocation: R1\_T8
  - Grup: G1
  - Aula: Q3-1009
  - Professor: -
- Date and Time fields:
  - Inici: 12 : 00 01 / 09 / 09
  - Fi: 13 : 00 01 / 09 / 09

At the bottom are "Aceptar" and "Cancelar" buttons.

Aquesta és la finestra de l'editor de nodes, per associar un grup d'alumnes a aquesta activitat només hem de fer clic en afegir per associar o eliminar per treure l'associació entre aquesta activitat i el grup d'alumnes. Finalment si només ens interessa modificar els recursos assignats hem de canviar la informació i fer clic en acceptar.

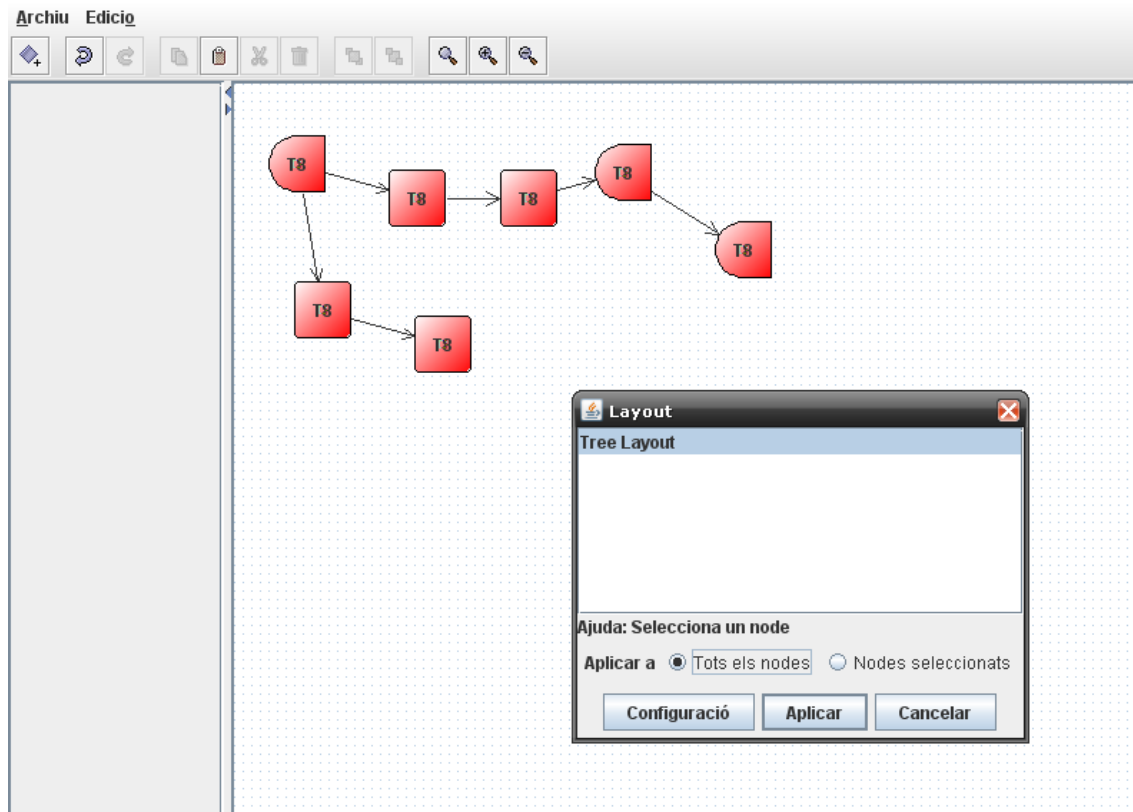
## 5.4 Ordenació topològica

Mitjançant l'opció Layout del menú Edició, podem accedir a l'ordenació topològica, per fer servir aquesta funcionalitat hi ha dues maneres, podem ordenar tot el graf o només els nodes que tinguem seleccionats.

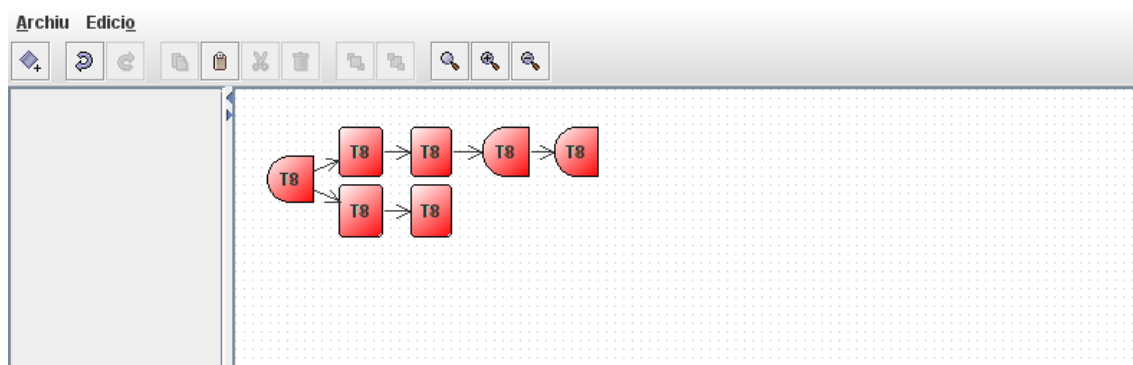
Al fer clic en aquest menú apareixerà una finestra semblant a la següent on podem triar el mètodes que volem fer servir, tots els nodes o només els seleccionats, podem triar

l'algoritme d'ordenació que volem i a més es pot canviar algunes propietats com la orientació d'ordenació, la distància entre nodes o la distància entre nivells.

Seguidament es pot veure un exemple d'utilització de l'ordenació topològica.

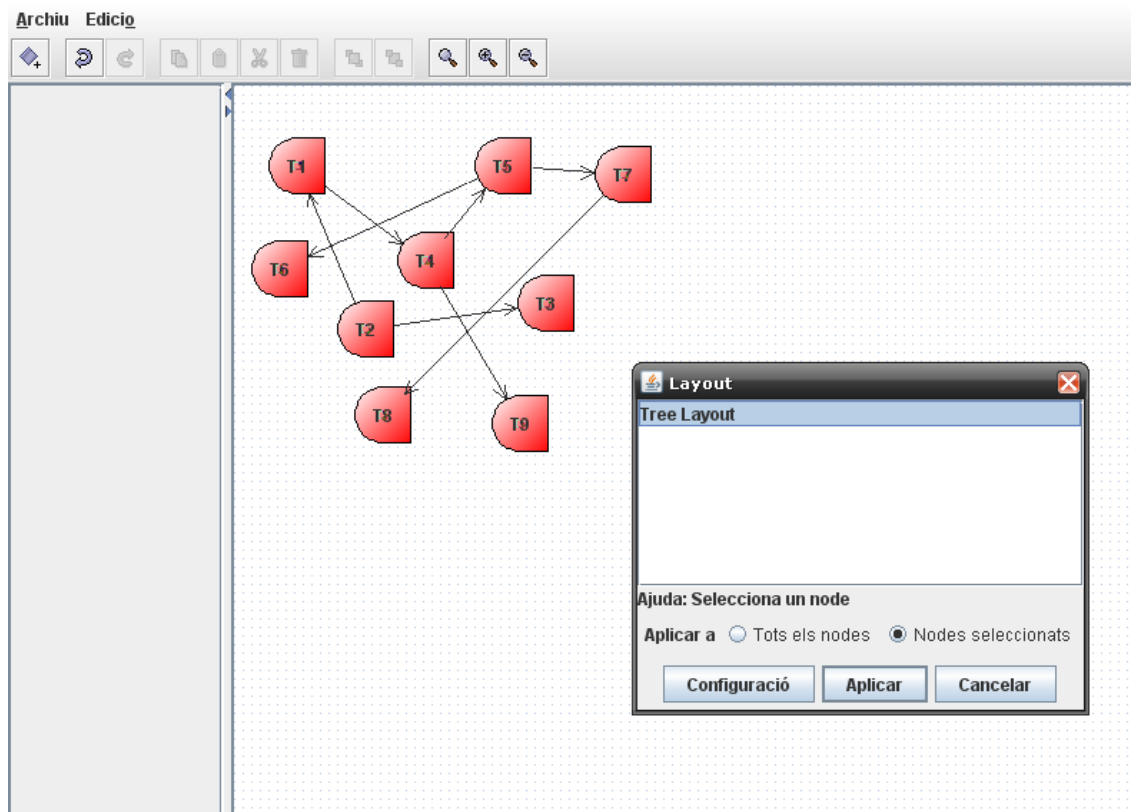


En aquest exemple tenim un graf desordenat i hem accedit a l'ordenació topològica, es pot veure com farem servir l'opció d'ordenar tot els nodes. Si apliquem aquestes opcions i fent servir el mètode Tree Layout, obtindrem un resultat semblant als següents.

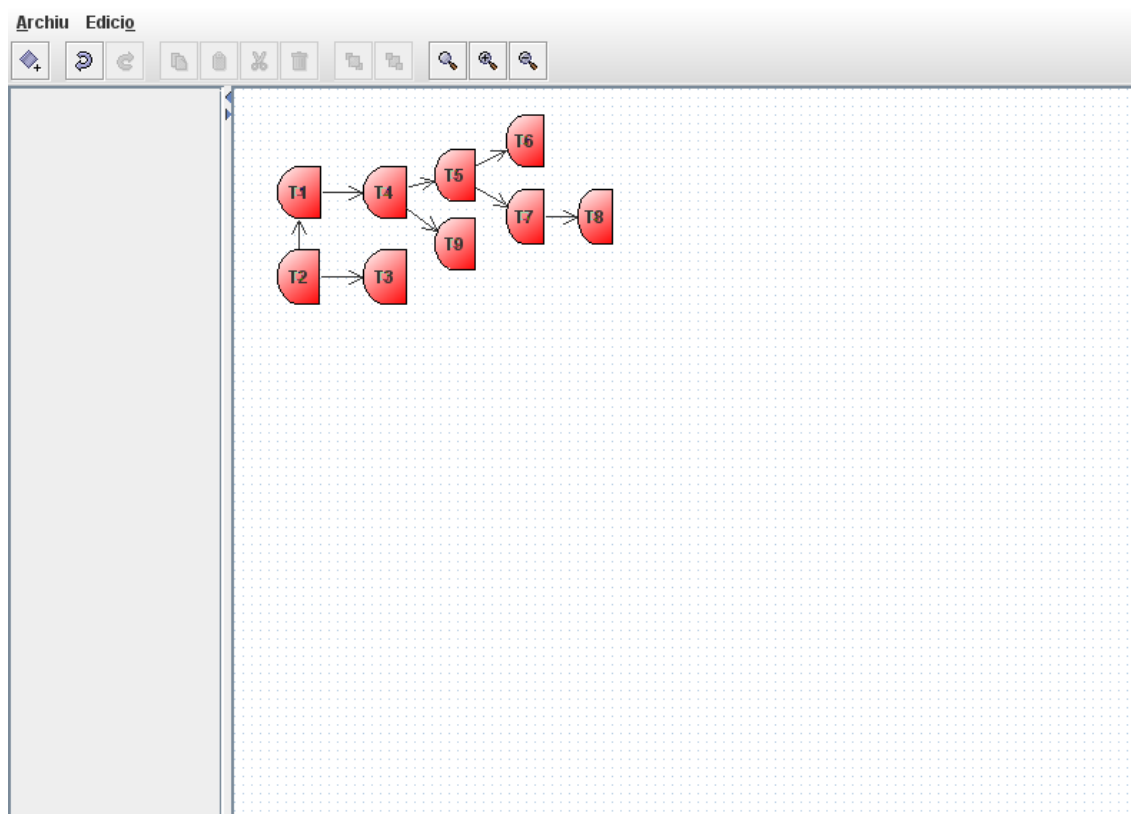




Finalment veurem un exemple on els arcs es creuen, al utilitzar l'ordenació topològica podem trobar uns resultats com els següents.



Després d'aplicar l'algoritme:



# Capítol 6.

## Descripció tècnica

Aquest capítol ha de servir com a guia per poder entendre el funcionament intern de l'aplicació per tal que qualsevol persona pugui ampliar l'aplicació en un futur.

Abans de començar a parlar sobre les classes Java que componen l'aplicació, un document molt útil per entendre com funciona la llibreria Jgraph és el Jgraph User Manual [1] on es descriu el funcionament de la llibreria i on es parla de conceptes tant importants com el Graph Model, el Graph Layout Cache i sobretot en parla de les cel·les, dels diferents tipus, dataobjects dels nodes les vistes i els renderers.

### 6.1 Estructura interna

La classe principal de l'aplicació és la Editor.java, en aquesta classe es crea la interfície gràfica i és on es troben la majoria de funcionalitats. En primer lloc es crida al constructor Editor() on es crea un nou graf cridant a la funció createGraf(), a més es creen els menús desplegable i s'instal·len els action listener que controlen tots els events que apareixen durant l'ús de l'aplicació.

A continuació trobem la funció insert(), aquesta funció és l'encarregada de crear un node nou, insertant-lo al layout i creant les seves propietats com la posició, color, tamany, etc.. A més es crea el dataobject del node que conté la informació del node a la funció createDefaultGraphCell().

Existeix una funció molt semblant a la insert què és la funció connect(). Aquesta funció el que fa és insertar un arc entre els ports que rep com a paràmetres. Cada node té un

port i en aquest port es on es connecta amb els altres nodes, a la figura 1 es pot veure un exemple. Semblant a quan s'inserta un node es creen les propietat i els atributs de l'arc .

Les següents funcions corresponen a les accions del menú d'eines com toFront que fa que el node seleccionat estigui al front, toBack que envia el node al fons i com si fos un sistema de capes sempre quedaria amagat per els altres nodes. Les funcions Undo, Redo i UpdateComandHistory corresponen al gestor d'accions per poder desfer i fer accions.

La funció creatToolBar() crea el menú d'eines i a més es defineixen les accions que han de fer cadascun dels botons definits a la barra d'eines.

Finalment trobem totes les funcions relacionades amb l'XML, com les funcions per serialitzar i deserialitzar les classes, una altra funció que inicialitza l'encoder de java per xml i una altra pel decoder. Aquesta part està perfectament explicada al manual del Jgraph a la pàgina 96.

## 6.2 Editor de nodes

Dintre de l'editor de graf existeix el que s'anomena celleditor, aquest celleditor no és més que un JComponent que conté un formulari on es carrega la informació del node i permet modificar i afegir informació. Per defecte la llibreria Jgraph només permet emmagatzemar als nodes un String i per tant l'editor només conté un textbox. En l'editor de guies docents cada node té molta més informació, per tant es necessari un editor molt més complex que ens permeti treballar amb les estructures de dades que té cada node.

A les classes NodeData, Allocation i Fecha tenim l'estructura de dades que conté cada dataobject i a la classe MyEditorComponent és on es crea el JComponent que se li passa al graf com a customEditor. Dins de la classe MyEditorComponent hi han varies funcions, com per exemple la funció initComponents(), en aquesta funció es crea el component principal que conté el formulari de l'editor, per inicialitzar els valors dels textfields existeix la funció installvalue(), aquesta funció el que fa es actualitzar els

valors dels textfields segons el dataobject que rep com a paràmetre. Finalment la funció `actionPerformed()` és l'encarregada d'actualitzar el dataobject amb les dades que hi tinguem en el formulari.

La classe `MyBasicUI` és l'encarregada del que passa abans i després d'editar un node. Aquesta classe crea i crida a l'editor amb les funcions `createEditDialog()` i `startEditing()` respectivament. Quan es pren el botó acceptar de l'editor de nodes es completa l'edició i per tant es crida a la funció `completeEditing()` d'aquesta classe. En aquesta funció es canvia l'atribut relacionat amb la forma que tindrà el node depenent del seu contingut, mitjançant el següent codi:

```
Map nested = new Hashtable();
Map map = new Hashtable();
map.put("renderKey", "tipo4");
nested.put(oldCell,map);
```

Aquest codi el que fa es canviar l'atribut `renderKey` que és el que conté la forma que tindrà el node a la forma "tipo4". Cal recordar que existeixen 4 fenotips diferents.

## 6.3 Vista dels nodes

En primer lloc a la classe `Editor` definim quina serà la `viewFactory`, en aquest cas serà una factory personalitzada com és la classe `GPCellViewFactory`, quan inicialitzem la `viewFactory` li hem de passar quin serà el renderer encarregat de dibuixar les vistes. Serà la classe `MyRendererView`, aquest renderer és un renderer universal, que mitjançant un switch a la funció `paint` dibuixarà una representació o una altra depenent de l'atribut `renderKey`.

A part de la funció `paint()` també hi ha una altra funció important què és la `getPerimeterPoint()`. Aquesta funció està relacionada amb els arcs i calcula des de on surten els arcs i fins a quin punt arriben, o sigui que controla que les fletxes dels arcs arribin just fins al perímetre dels nodes i no quedin dins o separats dels nodes.

## 6.4 Ordenació del graf

La classe `LayoutDialog` crea el component que conté el menú per configurar i aplicar l'ordenació del graf. Hi podem tenir tants algoritmes com vulguem, aquest algoritmes han de tenir dos classes una que implementi la classe `JGraphLayoutSettings` que conté la configuració i una altra que implementi la classe `JGraphLayoutAlgorithm` que conté l'algoritme d'ordenació.

## Referències

- [1] “Jgraph and Jgraph Layout Pro User Manual”, 14/09/09,  
[[www.jgraph.com/pub/jgraphmanual.pdf](http://www.jgraph.com/pub/jgraphmanual.pdf)]

# Capítol 7.

## Conclusions

L'objectiu d'aquest projecte és aconseguir una eina o intentar demostrar que es pot arribar a una aplicació per a l'edició de les agendes d'aprenentatge. Tot i que aquest projecte pot no tenir totes les funcionalitats de què disposa una eina comercial o una de lliure, que permeten l'edició de grafs en les seves versions finals, com ara la possibilitat de tenir una biblioteca de nodes on s'hi guarden per facilitar l'edició, tenir una vista global dels grafs, permetre l'exportació a diferents i altres opcions, l'eina del projecte ofereix totes les funcionalitats bàsiques per a l'edició de guies docents. D'aquesta manera, servirà com a base per una possible aplicació final i que es podrà utilitzar en casos reals com per exemple la planificació de qualsevol assignatura que segueixi els criteris d'ECTS.

Podem parlar del projecte com una prova de concepte per veure si totes les idees que hi havien en un principi abans de començar el projecte es podien dur a terme i que finalment es pot veure com que si que és factible un editor de guies docents amb els requeriments proposats en un principi.

Per poder veure si el projecte es podria considerar com una aplicació final i no com una prova de concepte en un principi s'hauria d'haver fet una prova amb un cas real on realment es veurien les limitacions de l'aplicació i a partir de la qual es podria ajustar el funcionament de l'aplicació fins a arribar als resultats que es desitgen.

Els resultats són una aplicació de demostració de viabilitat i proves de funcionament. Per tant, al no existir aquesta prova amb una planificació real d'una assignatura, és a dir, treballar amb un graf que pot superar els 200 nodes, caldria fer-ho amb un cas real per veure aquesta viabilitat ja que tot i que per casos de pocs nodes funciona perfectament

no sabem com es comportarà per aquests casos amb molt nodes. Es considera com resultats fer una prova per veure si el resultat que obtenim correspon amb tenir una aplicació que com es parla en els primers capítols, és fàcil d'utilitzar, permet a un usuari que no està familiaritzat amb aquest tipus d'eines utilitzar-la sense cap tipus de problema i principalment que permeti l'edició de guies docents. Per tant no parlar de resultats no vol dir que l'aplicació no funcioni correctament ni que no es compleixin els requisits sinó que només s'han fet proves amb casos amb pocs nodes.

La motivació d'aquest projecte era la manca d'una eina que ens permetés realitzar una planificació però que al mateix temps es pogués gestionar els continguts de manera fàcil, mitjançant l'utilització de les formes com a representació dels continguts s'ha aconseguit un editor de grafs que a més de permetre fer una planificació i gestionar els continguts dels nodes, aconsegueix l'objectiu de poder tractar els continguts dels nodes de manera molt fàcil ja que visualment es pot saber quins atributs hi ha definits al nodes.

Si analitzem la planificació es pot veure com no s'ha complert la planificació, es pot dir que era massa optimista en un principi i per manca de temps i la limitació de les 450 hores hi han algunes tasques que no s'han pogut realitzar. A més algunes tasques es van pensar amb una durada i finalment la quantitat d'hores necessàries per acabar aquesta tasca ha estat molt més gran. Aquest error en la planificació penso que pot ser degut a la inexperiència fent planificacions ja que aquesta planificació era la primera planificació d'un projecte real.

De cara al futur ja s'ha dit que aquesta aplicació podria ser la base per una aplicació per treballar en casos reals, o sigui que es podria utilitzar per planificar les assignatures de cursos futurs. De cara a aquesta aplicació es podrien incorporar algunes funcionalitats com per exemple mostrar més informació de forma visual aprofitant el canvi de forma que ja existeix, ara mateix el canvi de forma serveix per indicar que tenim una activitat amb la seva descripció, activitats associades a grups d'alumnes (amb una de les vores arrodonida) i per activitats amb recursos associats (l'altra vora arrodonida), un dels canvis seria que quan tenim més d'un grup d'alumnes el node amb la vora arrodonida tingués tantes capes com grups en tingués assignada l'activitat. I a més fent clic en cadascuna de les capes es pogués accedir a l'informació de cadascun dels grups.

Una altra ampliació que també està relacionada amb proporcionar informació de forma visual seria la de fer que els nodes poguessin tenir una llargada proporcional al temps de durada de l'activitat, sempre que aquesta en tingués una durada assignada, sinó en tindria una llargada mínima per defecte.

De cara al punt de vista dels alumnes també seria interessant que al accedir poguessin modificar algunes propietats dels nodes com el color per indicar més prioritat per exemple i que es poguessin visualitzar les planificacions de diferents assignatures a la vegada per planificar-ne el curs millor.





A partir de la implantació de ECTS en l'àmbit de l'EEES, la nova metodologia docent, ha augmentat la importància de les guies docents. Això és degut a que l'alumne passa a ser l'actor principal del procés d'aprenentatge. Això provoca la creació d'agendes d'aprenentatge però per poder oferir als alumnes aquestes agendas es crea la necessitat de disposar d'una eina que ens permeti introduir les guies docents de manera que tinguem una integració entre agenda i guia docent. En aquest projecte s'intenta aconseguir aquesta eina que a més de permetre introduir les guies docents, ens permeti obtenir informació de manera visual sobre les activitats d'aquesta guia docent.

A partir de la implantación de ECTS en el ámbito del EEES, la nueva metodología docente, ha aumentado la importancia de las guías docentes. Esto es debido a que el alumno pasa a ser el actor principal del proceso de aprendizaje. Además provoca la creación de agendas de aprendizaje pero para poder ofrecerlas a los alumnos, se crea la necesidad de disponer de una herramienta que nos permita introducir las guías docentes de manera que tengamos una integración entre agenda i guía docente. En este proyecto se intenta conseguir esta herramienta que además de permitir introducir las guías docentes, nos proporcione información de forma visual acerca de las actividades de la guía docente.

About the implementation of the ECTS in the ambit of EEES, the new teaching methodology has increased the importance of the educational guides. This is due to the student begun to be the main actor at the learning process. Also causes the creation of the learning agendas but for offer to the students, it makes the need of a software that allows us to make the learning guides by integration of agenda and educational guides. In this project, I try to reach this software that besides allow the introduction of the educational guides, provides visual information about the activities of the educational guide.